

REMARKS

Please reconsider the application in view of the above amendments and the following remarks. Applicant thanks the Examiner for courtesies extended during the Examiner Interview of June 6, 2005.

Disposition of Claims

Claims 1-8 are currently pending in this application. Claim 3 has been canceled by this reply. Claim 9 has been added by this reply. Claims 1 and 9 are independent. Claims 2, and 4-8 depend, directly or indirectly, from claim 1.

Drawings

Applicant respectfully requests the Examiner to accept the replacement drawings filed on November 15, 2004.

Rejections under 35 U.S.C. § 103

Claims 1-8 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,173,440 ("Darty") in view of U.S. Patent No. 6,513,154 ("Porterfield"). Claim 1 has been amended by this reply to further define the phrase "bug list" and specify that a minimal module is appended to the bug list. Further, the subject matter of dependent claim 3 has been amended into claim 1 to further clarify the concept of decomposing software modules. Support for the amendments made to independent claim 1 may be found, for example, on pages 6, lines 3-4, Figures 4-5, page 10, lines 14-20, and page 16. Support for newly added claim 9 may be found on page 18 of the Specification. To the extent that this rejection may still apply to the amended claims, this rejection is respectfully traversed.

The claimed invention is directed toward debugging software by determining minimal software modules of code by applying decomposition tests, isolating software bugs in the minimal software modules, and determining a minimal input vector, that when applied to the minimal software module, results in obtaining the software bug (*See Specification, page 6, lines 11-13*). Specifically, a software module is decomposed, by applying a module decomposition test, into smaller and smaller submodules. More specifically, the submodules are placed on a module decomposition list, where the module decomposition list is iteratively processed to further decompose (*i.e., remove*) submodules until a minimal software module is obtained.

Further, the claimed invention discloses appending the minimal input vector and minimal software module to a bug list. The bug list includes a minimal set of stimuli (*i.e., software module and input test vector*) that can be used to *reproduce a software bug* in the software module (*See Specification, page 6, lines 3-4*). Applicant asserts that the bug list is an important feature of the claimed invention because the bug list allows a user or developer to easily identify the smallest module of code and the smallest input vector that reproduces a software bug.

Turning to the rejection of the claims, the Examiner admits that Darty fails to explicitly disclose obtaining a bug list including stimuli to reproduce a software bug in the software module, and appending the software module and first input vector to the bug list. Further, Darty fails to disclose removing submodules from a module decomposition list, as recited in amended independent claim 1, and as discussed during the Examiner Interview of June 6, 2005. In addition, Darty fails to disclose or suggest removing submodules to obtain a *minimal* module (*i.e., the smallest number of lines of codes*), which is subsequently appended to the bug list of the claimed invention. (*See Specification, page 6, line 8*).

However, the Examiner states that the test point mapping matrix disclosed in Darty suggests creating a test list where each entry is composed of a code section and the corresponding test being performed on it. The Examiner states that creating such a code and test/input mapping is taught by the bug list of Porterfield, and that it would have been obvious for one of ordinary skill in the art to create the mapping matrix so that it lists a mapping of code and input vector.

Applicant respectfully disagrees with the Examiner's assertions. Darty simply discloses mapping test points of modules into different matrices, and analyzing the test points repeatedly, to determine whether failures exist. However, as discussed during the Examiner Interview of June 6, 2005, Darty clearly does not disclose decomposing (*i.e.*, removing) submodules on a module decomposition list based on a module decomposition test. Further, Porterfield fails to supply that which Darty lacks. The cited portion of Porterfield (*i.e.*, Steps 34-39, Fig. 5) does not disclose decomposition of software modules into submodules, which are then decomposed further to obtain a minimal module that is appended to a bug list as defined above. Rather, Porterfield discloses that for each module in the target code, a "module task" is created in the Gantt chart. The Gantt chart is a planning document. Every module task iterates through its constituent procedures, creating procedure tasks on the planning document. All procedures iterate through automatic bugs, which occur in the procedure. If the bug is fixed, the task is marked as 100% completed. Automatic bugs can be individually attached to the Gantt chart.

Although the Gantt chart may include software bugs, the Gantt chart is not equivalent to the bug list of the claimed invention because the bug list of the claimed invention includes both minimal input vectors and minimal modules, where applying the input vector to the minimal module results in a software bug. However, the Gantt chart of Porterfield does not include

minimal modules (*i.e.*, modules decomposed into the smallest possible software module that results in a software bug). Further, because Porterfield does not disclose or suggest decomposing modules, it is not possible for the Gantt chart of Porterfield to be updated (*i.e.*, for items to be appended) when a minimal module has been determined.

Further, the Examiner asserts that branching back to S102 in Darty discloses applying a module decomposition test to a module when the result of the comparison test is positive. However, returning to S102 in Figure 3A from Figure 3D is explained in Darty as “the process branches back to step S102 and the process repeats, until the fault isolation set matrix is empty at the inspection step S153.” Thus, it is clear that returning to repeat the process disclosed in Darty is simply a check to ensure that the modifications made based on the failed test point in Darty results in the passing of all test points. There is no decomposition test applied in Darty when the results of a comparison between the expected values of a module of code and the actual values of the module of code show that the values are different. Therefore, although the fault isolation matrix of Darty isolates the functional block of code causing a test point failure, the functional block of code is not decomposed as required by the claimed invention.

Rather, in Darty, the functional blocks of code are defined by breaking of lines of code into functional blocks before the test points are tested. The functional blocks, once defined, are not minimized or decomposed (*i.e.*, nothing is removed from the functional blocks to minimize them). To the contrary, only matrices are used to isolate the functional block that is causing a test point error. This is completely different than the method of the claimed invention, where software modules are decomposed by applying decomposition tests to obtain a minimal module and a minimal test input vector that, when combined, result in a software bug. Thus, Darty fails to disclose the limitations “applying a *module decomposition test* to said module when the result

of the comparison is positive,” and “obtaining a *module decomposition list* comprising two or more submodules of said software module and iteratively processing said module decomposition list when the result of said module decomposition test is positive, wherein iteratively processing said module decomposition list comprises *removing* at least one submodule.”


In view of the above, it is clear that Darty and Porterfield, whether considered separately or in combination, fail to render amended claim 1 of the present invention obvious. Thus, claim 1 is patentable over Darty and Porterfield. Dependent claims 2-8 are patentable for at least the same reasons. Further, independent claim 9 includes the same limitations as independent claim 1 and is therefore patentable over Darty and Porterfield for at least the same reasons. Accordingly, withdrawal of this rejection is respectfully requested.

Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and places this application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 16159/062002).

Dated: June 10, 2005

Respectfully submitted,

By 
Robert P. Lord
Registration No.: 46,479
OSHA · LIANG LLP
1221 McKinney St., Suite 2800
Houston, Texas 77010
(713) 228-8600
(713) 228-8778 (Fax)
Attorney for Applicant